ABSTRACT OF THE DISCLOSURE

A system and method for dynamic scheduling of Hard Functions (i.e., code segments which are targeted and prepared for hardware execution in a software program) for execution in a Reconfigurable Logic Fabric Hardware Partition of a reconfigurable computing system is described. Prior to execution of a software program, specialize statements are inserted into the software program code to identify the beginning and end of each Function Call. In addition, specialize statements are inserted at the beginning and end of each Hard Function within the Function Call. The software program is compiled for both a soft execution and a hard execution. In addition, in the case in which Hard Functions are identified, Configuration Bits are generated which when loaded into a Configuration Memory are used to configure the reconfigurable logic fabric partitions so as to perform the function as defined by the Hard Function code segments. In soft execution, the Function Call is executed wholly in the system's embedded processor. In hard execution, the reconfigurable logic fabric performs the function of the Hard Function and code other than the Hard Function is performed by the embedded processor. A hierarchy of memory storage devices with incremental access latency is employed to stage the Configuration Bits of individual Hard Functions to arrive at a Configuration Memory in time when a function is called. Configuration Bits in the form of Virtual Partitions are moved or staged within the hierarchy of storage devices in pace with their estimated time of need. A Function Call History Model including statistical run characteristics is established for determining whether to activate soft execution or hard execution of each Hard Function. The Model is adaptively updated in real-time execution of each Function Call. The model allows the system to learn from its previous hits and misses dynamically. Also included in the Call History Model is statistical data reflecting the execution trends of a Function Call, in terms of frequency of use, performance gain, and execution duration. The statistical data is used to estimate time of need and expected gain for the staging of Virtual Partitions through the hierarchy of memory storage devices.

The system and method also includes a Training Mode in which the system starts up without an initial Call History Model and constructs such a Model on-the-fly. The adaptive nature of the present invention makes the self-training possible, and thus simplifies user interaction with the system.

5